

REMARKS AND ARGUMENTS

By way of an Office Action dated November 29, 2004, claims 1 through 38 were rejected under 35 U.S.C. §§ 112 and 102.

Concerning the rejections under 35 U.S.C. § 112, the Office Action assumed, for purposes of examination, that “said target application “was meant to be “said target software application”; “said executing software application” was meant to be “said target software application”; said “first grain” of claim 12 was meant to be “said first version grain”; and “said active grain” of claim 25 was meant to be “said active crumb”. Applicant has adopted these assumptions and amended the claims accordingly as shown below.

Concerning the rejection pursuant to 35 U.S.C. § 102, the Office Action solely relies upon U.S. Letters Patent 5,359,730 (hereinafter “Marron”) to anticipate claims 1 through 38. Herein, Applicant will illustrate why Marron does not anticipate claims 1 through 35 since Marron is limited to **replacement** of **entire** operating system modules after the operating system module has completed execution in mainframe processing, rather than for modification of existing software applications while they are executing without halting their execution. Specifically, Marron is directed to a system with “a plurality of central processing units” (Marron, col. 6, line 4-5) and “preferably a large commercially available IBM MVS/ESA [main frame] ... classed as a multitasking, multiprocessing [data processing system].” (Marron, col. 6, lines 9-13).

1. Marron does not anticipate "first version grains."

The Office Action states that Marron anticipates "first version grains" by reference to "change modules." (Office Action page 3, ¶ 10). However, Marron does not segment an existing software application into grains as in the pending application, but Marron contemplates and discloses the replacement of entire modules and programs. Marron does not claim or disclose the segmenting of an existing program into grains or crumbs. For example, Marron contains the following disclosures:

"The general problem that the invention addresses and solves is how to **replace** modules" (Marron, col. 6, lines 26-28). The invention dynamically installs "new modules and programs." (Marron, col. 6, line 55). The invention of Marron determines when to "stop executing the old program and start executing the new program." (Marron, col. 7, lines 5-7). Marron also "loads new copies or the new programs A' and B'." (Marron, col. 7, lines 55-56). Specifically, Marron requires that the programmer who wishes to implement changes to a mainframe operating system must create new programs, recompile the new programs, and link the new programs to form new programs. (Marron, col. 6, lines 50-54). Therefore, Marron does not disclose or anticipate claims 1 through 38 since these claims contemplate the segmentation of an existing target software application into grains for modification, rather than the entire replacement of programs or modules.

2 Marron does not anticipate the hot pack since "new modules" as stated in Marron do not contain grains.

The Office Action states that Marron anticipate the hot pack through its language of "new programs are created". (Office Action, page 3, ¶ 10). However, Marron state that the new program is created for replacement of the old program existing on the mainframe operating system. Specifically, Marron states that "the new programs are created by change programmer modifying the old programs, recompiling, and linking the new programs to form load modules." (Marron, col. 6, lines 50-54. The load modules contain change instructions for "dynamically installing the new modules and programs." (Marron, col. 6, line 55). The invention of the present application, however, directs modification to the target software application through replacement grains targeted to the target software application contained in the hot pack, rather than replacement of programs from old programs to new programs as disclosed in Marron. Specifically, Claim 1 states, "modifying at least one of said first version grains of said target software application"; Claim 8 states, "modifying at least one of said first version grains"; Claim 14 states, "a means for modifying at least one of said first version grains"; Claim 20 states, "modifying at least one of said first version grains of said target software application"; Claim 26 states, "modifying said first version grain associated with said dictum"; and Claim 32 states, "modifying at least one of said first version grains of said target software application." Therefore,

Marron does not anticipate a hot pack since Marron is limited to replacement of new programs or modules and does not claim or disclose modification of first version grains of a target software application.

3. Marron's use of the terms "old program" and "new program" does not anticipate first or second version grains.

The Office Action states that Marron anticipates "first version grains" through the disclosure of "modifying old programs." (Office Action, page 4, ¶ 10). Further, the Office Action states that Marron anticipates "second version grains" through the disclosure of "new programs." (Office Action, page 3, ¶ 10).

"First version grains" are segments of the target software application and are not entire old programs. Specifically, the pending application contains the following: "Grains delineate the source code and object code into discreet segments...." Further, "modifying the old programs", as stated in Marron, refers to the creation of the new program from the old program; a task performed by the programmer prior to modification on the mainframe operating system. (Marron, col. 6, lines 55-56). These new programs are complete programs or modules so that the old programs or modules are replaced. (Marron, col. 6, lines 21-26). The phrases "new program" and "old program" do not anticipate "first version grains" and "second version grains" since Marron does not claim or disclose segmenting a target software application into grains.

“New programs” as disclosed in Marron are complete, compiled, and linked executable programs that are not segments of programs (or grains as in the present invention) so that Marron does not anticipate second version grains. Specifically, Marron states that the purpose of the invention is “replacing old operating system programs or modules with new updated version....” (Marron, Abstract). Further, the invention of Marron must determine when to execute the old programs or when to execute the new programs. (Marron, Abstract and Fig 2B, step 64 “Switch Over To New Programs” and Fig 3, steps 72, “Route to New Program” and step 74, “Route to Old Program”).

A “second version grain” of the present invention is a segment of the target software application and does not execute independent of the target software application. Therefore, Marron does not claim or disclose a second version grain, only the new program or module. The modification of a first grain according to a second grain is not claimed or disclosed in Marron.

4. Marron does not claim or disclosure suspending execution of the target application.

The Office Action states that Marron’s use of the phrase “placing the process in a wait state” anticipates the suspension of the target software application. (Office Action, page 4, ¶ 10). However, the language of Marron does not discuss the target software application, but refers to the process that would be calling or executing the

target software application. Since Marron is designed for the mainframe, multiprocessor operating system, Marron is concerned with the replacement of operating system modules and programs in their entirety. Once Marron makes a replacement, the operating system needs to know if it is "safe" to execute the new program or not. (Marron, col. 7, lines 3-7 "the conditions which must be satisfied when processes can stop executing the old program start executing the new program"; col 7, lines 28-31, "any task attempting to execute the old code will be routed either to the old code if the process is unsafe or to the new code if the process is safe"). Therefore, this "wait" state of Marron means that the process that is to call the old or new program and must wait to see if it is "safe" to call the new program. Marron does not claim or disclose the suspension of the target software application so that the first version grains can be modified according to second version grains.

5. Marron does not claim or disclose the modification of a first version grain to a second version grain according to a dictum.

The Office Action states that "determining the status of said dictum" is anticipated by Marron's language, "to determine ... whether program A or program A' should be executed." In Marron, the new program A' needs to have been created, compiled, installed and existing on the mainframe multiprocessor operating system prior to the determination as to whether program A or program A' can be executed. Marron merely explains that a decision must be made as whether to execute program

A or A' (old or new). This language does not anticipate the determination as to whether to modify the first version grain according to a second version grain. Further, the dictum of the invention of the present application determines when and how the modification of the first version grain should occur. The dictum of the present invention does not determine which of two simultaneously existing programs (A and A') should be executed. Therefore, Marron does not anticipate the determining of the status of a dictum to determine whether to modify the first version grain according to a second version grain as stated in Claims 1, 8, 14, 20, 26, and 32 (all the independent claims).

6. Marron does not anticipate modification of a target software application by suspending of the target software application.

The Office Action states that Marron anticipates suspending the target software application by Marron's language "pass control to the ... program for execution". (Office Action, page 4, ¶ 10). However, this language in Marron refers to a program of the mainframe operating system encountering a null (0x00) and executing a standard program check first level interrupt handler (PCFLH). (Marron, col. 8, lines 20-25. Once PCFLH of Marron is executed, control is passed to the dynamic software update facility (DSUF). (Marron, col 8, lines 25-26). DSUF then determines whether the old program A or the new program A' should be executed. Marron, col. 8, lines 32-37). Marron only states that if the interrupt is not for the DSUF, then the PCFLH continues

to execute. Neither the DSUF or the PCFLH can be considered the target software application of this invention. Marron only states that control is not passed from PCFLH to the DSUF if the trap encountered is not a DSUF trap. Therefore, Marron does not claim or disclose the suspension of the target software application for modification of the first version grain.

7. Marron's "safety points" do not anticipate crumbs.

Marron's description of "safety points" does not claim or disclose crumbs of the present application. Safety points are defined in Marron as "conditions [that] can be translated into events in the life of a task, or the combination of an event with an observable state of the process." (Marron, col. 7, lines 7-9). Crumbs, however, are physical locations within a grain that can be used to determine when the execution pointer hits a predetermined point in the execution of the target software application. Marron specifically describes safety points as "when a task is: started after the change was implemented, ...entering or exiting a particular module, ...making a particular system call, executing an instruction at a given offset at a given module...observed as being in a problem or user state...observed as being in a wait state...running under a job name, and not running under a given job name." (Marron, col. 7, lines 12-24). Further, safety points are "either observed by the system since they include a system call, or observed by the DSUF...." (Marron, col. 7, lines 25-27). Marron's "safety points" do not anticipate crumbs because crumbs are locations in the target software

application that can allow dictums to be evaluated once reached by the execution point. Therefore, the "safety points" of Marron does not anticipate crumbs.

Although the arguments above are principally directed to the independent claims, they are equally applicable to the remaining dependent claims. For the reasons presented herein, Applicant respectfully requests that this application be granted a notice of allowance for claims 1 through 38 in the normal course of Patent Office business.

In the event that the Examiner does not find these amendments and arguments persuasive, the Applicant requests an interview to discuss the Office Action and claims of this pending application.

Respectfully submitted,



Douglas W. Kim
Registration No. 44,848
McNAIR LAW FIRM, P.A.
P.O. Box 10827
Greenville, SC 29603-0827
Telephone: 864-232-4261
Attorneys for the Applicant